



HAL
open science

Calage de paramètres hydrauliques pour des modèles de transport améliorés dans un réseau d'eau

L. Roulant

► **To cite this version:**

L. Roulant. Calage de paramètres hydrauliques pour des modèles de transport améliorés dans un réseau d'eau. Sciences de l'environnement. 2015. hal-02601874


HAL Id: hal-02601874

<https://hal.inrae.fr/hal-02601874>

Submitted on 16 May 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Calage de paramètres hydrauliques pour des modèles de transport améliorés dans un réseau d'eau.

Rapport de stage de 2ème année ENSEIRB-MATMECA

24 Septembre 2015

Loïs ROULANT
Promotion 2016

Responsable de stage :
Olivier PILLER

Tuteur universitaire :
Nathalie BONNETON



Table des matières

1	Présentation	2
1.1	Introduction	2
1.2	Remerciements	2
1.3	Irstea	2
1.4	<i>Smart – Online</i> ^{WDN}	3
2	L’analyse hydraulique	3
2.1	Equations d’équilibrage hydraulique	3
2.2	Problème inverse et calage	4
2.3	Code Matlab et modifications	4
3	Travaux réalisés	5
3.1	Lecture des mesures et prise en compte des pompes et de leur régulation	5
3.2	Ajout de la formule de perte de charge de Darcy-Weibach	5
3.3	Introduction de la PDM	7
3.4	Transport	9
4	Résultats et conclusion	11
4.1	Résultats	11
4.2	Conclusion	14

1 Présentation

1.1 Introduction

Le logiciel Porteau, distribué par l'organisme de recherche Irstea, est un logiciel qui permet la modélisation de réseaux maillés de distribution ou de transport d'eau sous pression. Ce logiciel est développé par l'équipe Gestion Patrimoniale des Infrastructures liées à l'Eau (GPIE) qui m'a accueilli pour ce stage. Les recherches menées au sein de cette équipe s'organisent autour de trois axes : l'évaluation et la maîtrise de la performance et des risques liés aux infrastructures, l'optimisation des stratégies de gestion patrimoniale dans le cadre d'une vision à long terme et la conception et le développement de méthodes et d'outils opérationnels. C'est dans cette dernière branche que s'inscrit mon stage puisqu'il visait essentiellement à développer le code Matlab du logiciel Porteau afin d'y implémenter de nouvelles fonctionnalités. Les travaux effectués dans ce stage sont en liaison avec des recherches actuelles de l'Irstea et de ses partenaires. Mes travaux ont été dirigés suivant trois axes principaux : l'amélioration du code Matlab existant avec par exemple la prise en compte de formules plus proches de la réalité ou le calage de nouveaux paramètres, l'utilisation d'un modèle avec des consommations dépendantes de la pression ou encore le calage de paramètres hydrauliques à partir de mesures de concentration. Tout d'abord, une présentation de l'Irstea et de ses différentes unités sera faite. Ensuite, nous exposerons ce qui a servi de base à mon travail avec notamment le modèle hydraulique utilisé, la formulation du problème de l'analyse et du problème inverse ainsi que la présentation du code Matlab existant. Ensuite, nous nous intéresserons aux travaux réalisés avec la lecture de mesures rentrées directement depuis l'interface du logiciel et l'implémentation d'une nouvelle formule de perte de charge ainsi que la construction des jacobiniennes associées. On verra également l'introduction de la PDM (Pression Dependant Model) dans le code Matlab ainsi que le transport et le calage associé. Pour finir, on présentera les résultats obtenus et une analyse de ces derniers sera faite.

1.2 Remerciements

Premièrement, je tiens à remercier particulièrement Olivier Piller, mon maître de stage, et Hervé Ung, en thèse à l'Irstea et très présent durant mon stage, qui m'ont permis de tirer profit au maximum de ce stage en partageant leurs connaissances et en prenant le temps de me transmettre au mieux leurs compétences dans des conditions optimales pour un stagiaire autant au niveau matériel qu'au niveau de l'environnement de travail. J'aimerais également remercier mes collègues de l'équipe GPIE qui auront su m'apporter de l'aide sur le plan scientifique et de la bonne humeur mais qui m'ont aussi permis de découvrir la vie en entreprise.

1.3 Irstea

L'Irstea, Institut national de Recherche en Sciences et Technologies pour l'Environnement et l'Agriculture a été créé en 1981. Originellement appelé Cemagref (Centre national du Machinisme Agricole, du Génie Rural et des Eaux et Forêts), l'Irstea est impliqué dans des domaines scientifiques très larges comme l'agriculture responsable et l'aménagement durable des territoires, la gestion de l'eau et les risques associés (Sécheresse, crues, inondations), l'étude des écosystèmes com-

plexes et de la biodiversité. L'Irstea est divisé en neuf centres de recherche situés à Aix-en-Provence, Antony, Bordeaux, Clermont-Ferrand, Grenoble, Lyon, Montpellier, Nogent-sur-Vernisson et Rennes. Le centre Irstea Bordeaux, localisé sur le site de Cestas-Gazinet, est focalisé sur deux domaines principaux : la gestion de l'eau et du fonctionnement des milieux aquatiques et l'interface entre eau et gestion des territoires. Pour cela, deux unités de recherche sont présentes sur ce centre : EABX (Ecosystèmes Aquatiques et changement globaux) et ETBX (Environnement, Territoires et infrastructures). Ces unités comprennent chacune plusieurs équipes et l'équipe GPIE, au sein de laquelle s'est déroulé mon stage, appartient à l'unité ETBX.

1.4 *Smart – Online*^{WDN}

Le projet *Smart – Online*^{WDN} est un projet de recherche franco-allemand qui a commencé en avril 2012. Il consiste à améliorer deux solutions logicielles, dont le logiciel Porteau, qui permettent de modéliser en temps réel des réseaux de distribution d'eau pour favoriser la sécurité et la fiabilité. En effet, les réseaux de distribution d'eau peuvent être sujets à des intrusions de contaminants à n'importe quel endroit du réseau. C'est pourquoi les chercheurs travaillant sur ce projet tentent, entre autres, de modéliser des prédictions fiables sur la quantité et la qualité de l'eau présente dans les réseaux de distribution pour permettre de remonter aux sources de contaminations. L'unité "ETBX" de l'Irstea effectue des travaux de recherche sur le fonctionnement hydraulique des systèmes de distribution d'eau potable, le vieillissement des réseaux et la maîtrise des pertes dans les réseaux. Des outils de diagnostic des réseaux comme le logiciel Porteau et d'analyse du vieillissement et du renouvellement sont développés et distribués. L'équipe GPIE, un groupe de cette unité, coordonne le projet Smart-Online et travaille en collaboration avec sept partenaires publics et privés franco-allemands. La finalité de ce projet permettra à certaines entreprises comme par exemple Véolia environnement, la CUS ou BWB d'utiliser ce logiciel pour placer de façon optimale des capteurs mesurant plusieurs paramètres physiques et chimiques de l'eau, établir un modèle de simulation fiable de la quantité et la qualité de l'eau dans ses réseaux, identifier la source de la contamination et analyser les risques et les impacts au sein de ses réseaux de distribution d'eau.

2 L'analyse hydraulique

2.1 Equations d'équilibrage hydraulique

Mon stage s'est composé de plusieurs phases, la première étant de lire la documentation proposée sur le sujet. Durant cette première partie, j'ai pu, grâce à la thèse de mon tuteur Olivier PILLER [1], découvrir la modélisation de réseaux d'eau ainsi que l'analyse hydraulique qui en découle. Je vais donc commencer par exposer ce qui servira de base pour la suite. Un réseau de distribution d'eau est un ensemble de conduites, de réservoirs, d'équipements, *i.e.* des éléments hydrauliques comme des pompes, des vannes, des limiteurs de débits, des stabilisateurs de pression ... et de singularités *i.e.* coudes, rétrécissements,... La topologie du réseau, les longueurs et diamètres des tronçons et les caractéristiques des équipements et des singularités sont connues. Un tronçon est formé par une ou plusieurs conduites identiques. Le réseau est constitué de a arcs ou tronçons et N nœuds qui sont les extrémités de ces

tronçons. L'ensemble des nœuds est divisé en f nœuds à charge fixée ou nœuds réservoirs (il y a toujours au moins un nœud réservoir par réseau) et n nœuds simples. Dans un premier temps, on cherche à établir les équations d'équilibrage hydraulique, qui vont donner les relations entre débit (q), charges (h), demandes (d) et résistances (r) dans le réseau. Pour cela, il faut définir la matrice d'incidence du réseau $A^N \in M_{N,a}$ définie par :

$$A_{i,j}^N = \begin{cases} 1 & \text{si l'arc } j \text{ est orienté sortant de } i \\ -1 & \text{si l'arc } j \text{ est orienté rentrant de } i \\ 0 & \text{sinon} \end{cases}$$

Cette matrice se décompose en deux sous-matrices A^f la matrice d'incidence nœud-arc réduite aux nœuds à charge fixe (nœuds réservoirs) et A la matrice d'incidence nœud-arc réduite aux nœuds simples. On a alors : $A^N = \begin{pmatrix} A \\ A^f \end{pmatrix}$ où $A \in M_{n,a}$ et $A^f \in M_{f,a}$.

On écrit alors les équations d'équilibrage hydraulique grâce à la conservation de la masse et celle de l'énergie :

$$\begin{cases} A \cdot q = -d & \text{aux nœuds simples} \\ \xi(r, q) = {}^t A \cdot h + {}^t A^f \cdot h^f \end{cases} \quad (1)$$

avec ξ la perte de charge.

La perte de charge s'exprime en mètre de colonne d'eau et est calculée sur chaque arc comme une somme des pertes de charges liées aux pompes, de celles liée aux singularités et de l'énergie perdue sous forme de chaleur. Cette dernière partie de la somme, notée $\xi^{lin}(r, q)$ dépend de la formule de perte de charge choisie. Pour l'instant, seule la formule d'Hazen-Williams était utilisée dans le code : $\xi^{lin}(r, q) = r \cdot q \cdot |q|^{0.852}$.

2.2 Problème inverse et calage

Dans cette partie, nous nous intéressons principalement au problème inverse de l'Analyse, c'est-à-dire que nous allons utiliser les équations d'équilibrage en se basant sur des mesures afin de déterminer les rugosités et les demandes. Une mesure systématique de tous les débits et de toutes les charges permettrait une résolution directe et facile de ce problème inverse. Cependant, le nombre de mesures peut devenir très important sur des réseaux à l'échelle d'une ville et entraîner des coûts trop élevés. L'intérêt du calage est alors mis en évidence : à partir de quelques mesures bien placées, nous allons déterminer les rugosités des arcs et les demandes aux nœuds. Pour cela, nous allons poser un vecteur d'état X estimant les valeurs inconnues (rugosités et demandes). Le problème du calage de paramètres hydrauliques consiste maintenant à trouver X tel que $y(X) = y^{mes}$ avec y^{mes} le vecteur des valeurs mesurées et $y(X)$ la valeur calculée de à partir de X en utilisant une méthode d'optimisation.

2.3 Code Matlab et modifications

La seconde phase du stage a été la prise en main du code Matlab existant et les premières modifications. Le code Matlab se compose de trois parties qui vont permettre de résoudre le problème de l'Analyse, le problème inverse et enfin une partie

qui concerne le transport. La partie transport correspond à l'étude de la qualité de l'eau (concentration en polluant, âge et provenance de l'eau) et sera la dernière étudiée. Concernant les deux premières parties, j'ai commencé par comprendre le code existant en testant les différents sous-programmes qui le composent. Une batterie de test réalisée avec différents réseaux m'a permis de mettre en évidence quelques erreurs qu'il a fallu corriger afin d'avoir un code opérationnel comme base pour mon stage. La compréhension du stage est également passée par la compréhension de l'algorithme de résolution du problème inverse grâce aux travaux [2] de Claire Latappy (ancienne stagiaire de l'équipe GPIE issue de l'Enseirb-Matmeca). Ce code Matlab permet actuellement le calage de la demande et de la pression sur de petits réseaux en utilisant un modèle de consommation indépendant de la pression et une formule de perte de charge d'Hazen-Williams. Les limites concernent une lecture des mesures assez complexe et peu efficace pour de grands réseaux et un modèle hydraulique et une formule de perte de charge simple mais peu représentatifs de la réalité.

3 Travaux réalisés

Après cette phase d'immersion dans le stage et de compréhension, il a fallu déterminer les travaux que j'allais devoir réaliser. Inscrits dans des recherches en coopération avec l'université d'Adélaïde [3], ils ont pour buts de répondre aux limites citées ci-dessus.

3.1 Lecture des mesures et prise en compte des pompes et de leur régulation

La première tâche qui m'a été confiée a été de simplifier la lecture des mesures pour la partie calage. En effet, le code Matlab ne permettait pas une lecture directe des mesures puisqu'il fallait un fichier texte avec une certaine syntaxe pour permettre l'intégration de mesures. L'objectif était ici de supprimer ce fichier texte et de pouvoir utiliser les mesures données par l'utilisateur dans l'interface du logiciel Porteau. Pour cela, il a fallu implémenter le chemin vers le code Java où il était possible d'avoir les informations souhaitées. Une fois les mesures collectées, il a fallu recréer une structure de données identique à celle qui était déjà utilisée afin de coller à la suite de code existant. A cette lecture, j'ai également dû ajouter la lecture d'équipements présents sur les tronçons (comme les pompes ou les vannes motorisées) et modifier le code afin de prendre en compte ces singularités (notamment dans le calcul des pertes de charges). Après des tests concluants utilisant la nouvelle lecture des mesures sur le calage du réseau Caltest (Annexe A.1), cette nouvelle fonctionnalité a été validée.

3.2 Ajout de la formule de perte de charge de Darcy-Weibach

La seconde tâche qui m'a été affectée fut celle de l'implémentation d'une nouvelle formule de perte de charge. Jusqu'alors, seule la formule simple d'Hazen-Williams était utilisée dans le code Matlab alors que le logiciel Porteau permet l'utilisation de la formule plus complexe de Darcy-Weibach. Ceci a été réalisé à l'aide des travaux [3] de Sylvan Elhay (collaborateur de mon tuteur).

Généralement, la formule de perte de charge s'écrit sous la forme $\xi(r, q) = r \cdot q \cdot |q|^n$. Dans le cas de la formule d'Hazen-Williams on retrouvait cette formulation

avec $n=0.852$ et $r = \frac{10.67 \cdot L}{\left(\frac{diam}{1000}\right)^{4.87} (1000 \cdot Chw)^{1.852}}$ où dans le cadre de la formule d'Hazen-Williams, la rugosité est un coefficient noté Chw sans dimension.

L'introduction de la formule de Darcy-Weibach va permettre d'adapter la perte de charge au régime rencontré.

On aura une première formule en régime laminaire ($Re \leq 2000$) : $\xi(r, q) = \frac{128 \cdot L \cdot \nu}{\pi \cdot g \cdot diam^4} \cdot q$ on pourra remarquer que cette formule est linéaire.

Le cas turbulent ($Re \geq 4000$) est lui aussi assez simple et nous avons la formule $\xi(r, q) = \frac{2 \cdot \ln^2(10) \cdot L}{\pi^2 \cdot g \cdot diam^5} \frac{1}{\ln^2(u)} \cdot q \cdot |q|$ La résistance dépend ici du Reynolds et donc de q .

Enfin, le cas du régime transitoire ($2000 \leq Re \leq 4000$) est lui un peu plus compliqué : $\xi(r, q) = \frac{8 \cdot L}{\pi^2 \cdot g \cdot diam^5} \sum_{k=0}^3 \left(\alpha_k + \frac{\beta_k}{u}\right) \eta^k \cdot q \cdot |q|$ avec description des grandeurs en annexe B.1

Une fois ce travail effectué, il a fallu regarder les utilisations faites de cette perte de charge. Or, dans l'algorithme de Levenberg-Marquardt utilisé dans le programme que l'on peut voir en annexe C.1, on utilise également la Jacobienne du vecteur y . La Jacobienne est calculée suivant l'écriture en annexe B.2 :

Il a donc fallu implémenter le calcul des grandeurs $\partial_{rug}q, \partial_dq, \partial_{diam}q, \partial_{rug}h, \partial_dh$ et $\partial_{diam}h$ suivant les différents régimes avec la formule de Darcy-Weibach (la référence [4] est largement utilisée dans la suite de ce développement).

Premièrement, on définit les égalités suivantes :

$$rug = K_C \cdot C_{rug}, d = K_d \cdot C_{dem} \text{ et } diam = K_\phi \cdot C_{diam} \quad (2)$$

où les K_C, K_ϕ et K_d sont les matrices de sélection et C_{rug}, C_{dem} et C_{diam} sont les vecteurs de classes de rugosité, demande et diamètre.

Le problème de l'analyse Eq.1 est modifié en injectant Eq.2 :

$$\begin{cases} A \cdot q = -K_d \cdot C_{dem} \\ \xi(q, r(K_C \cdot C_{rug})) - {}^t A \cdot h - {}^t A_f \cdot h_f = 0 \end{cases} \quad (3)$$

En dérivant par rapport à rug et d ce système et après résolution, on obtient :

$$\begin{cases} \partial_{rug}q = -D^{-1} \cdot B \cdot R_{rug} + D^{-1} \cdot {}^t A \cdot (A \cdot D^{-1} \cdot {}^t A)^{-1} \cdot A \cdot D^{-1} \cdot B \cdot R_{rug} \\ \partial_{rug}h = (A \cdot D^{-1} \cdot {}^t A)^{-1} \cdot A \cdot D^{-1} \cdot B \cdot R_{rug} \\ \partial_dq = -D^{-1} \cdot {}^t A \cdot (A \cdot D^{-1} \cdot {}^t A)^{-1} \\ \partial_dh = -(A \cdot D^{-1} \cdot {}^t A)^{-1} \end{cases} \quad (4)$$

où $D = \partial_d(\xi)$, $R_{rug} = \partial_{rug}(r)$ et $B = \partial_r(\xi)$

La nouveauté apportée est la dérivation par rapport aux diamètres :

$$\begin{cases} \partial_{diam}(A \cdot q) = -\partial_{diam}(K_d \cdot C_{dem}) \\ \partial_{diam}(\xi(q, r(K_C \cdot C_{rug}))) - \partial_{diam}({}^t A \cdot h - {}^t A_f \cdot h_f) = 0 \end{cases}$$

puis $\begin{cases} A \cdot \partial_{diam}(q) = 0 \\ \partial_q(\xi) \cdot \partial_{diam}(q) + \partial_r(\xi) \cdot \partial_{diam}(r) - {}^t A \cdot \partial_{diam}(h) = 0 \end{cases}$. On obtient alors finalement :

$$\begin{cases} \partial_{diam}q = -D^{-1} \cdot B \cdot R_\phi + D^{-1} \cdot {}^t A \cdot (A \cdot D^{-1} \cdot {}^t A)^{-1} \cdot A \cdot D^{-1} \cdot B \cdot R_\phi \\ \partial_{diam}h = (A \cdot D^{-1} \cdot {}^t A)^{-1} \cdot A \cdot D^{-1} \cdot B \cdot R_\phi \end{cases} \quad (5)$$

avec $R_\phi = \partial_{diam}(r)$. On a calculé (annexe D) les matrices D, B, R_{rug} et R_ϕ pour chaque régime dans le cas de la formule de Darcy-Weibach.

3.3 Introduction de la PDM

Principe

La PDM (Pression Dependant Model) est une adaptation des équations d'équilibre hydraulique, et plus particulièrement de la conservation de la masse. En effet, comme on peut le voir dans la première partie où ces équations d'équilibre sont établies, on considère qu'à chaque nœud, la somme des débits entrants et sortants est égale à la demande des consommateurs : c'est le modèle DDM (Demand Driven Model). Dans ce modèle parfait, la demande du client est alors toujours pleinement satisfaite et sa consommation (ce qu'il « prélève » du nœud) est égale à sa demande (ce qu'il aimerait y « prélever »). En réalité, la demande du client n'est pas toujours satisfaite sa consommation va dépendre de la pression au nœud, d'où le terme Pression Dépendant Model. On a alors une nouvelle formulation des équations d'équilibre hydrauliques où la demande d est remplacée par la consommation c qui dépend de la pression (h) et de la demande du client (d) :

$$\begin{cases} A \cdot q = -c(d, h) \\ \xi(r, q) - {}^t A \cdot h - {}^t A_f \cdot h_f = 0 \end{cases} \quad (6)$$

Il faut maintenant modéliser au mieux la consommation d'un client en fonction de sa demande et de la pression afin d'obtenir une fonction c exploitable. Plusieurs choix pour la fonction c sont disponibles et devront être sélectionnés dans le code. Ces différentes fonctions c utilisent toute une pression « réduite » ($hred$) calculé comme suit : $hred = \frac{h - CoteSol}{CoteDésirée - CoteSol}$ où la côte désirée représente la pression minimum pour satisfaire la demande du consommateur.

On présente une des trois fonction c implémentées dans le code (les deux autres sont disponibles en annexe B.3 :

- Régularisation de la Fonction de Wagner régularisée [7] [8]

$$c(d, h) \begin{cases} d & si \ hred \geq 1 + \varepsilon \\ d \cdot (a_2 \cdot hred^3 + b_2 \cdot hred^2 + c_2 \cdot hred + d_2) & si \ 1 - \varepsilon \leq hred \leq 1 + \varepsilon \\ d \cdot \sqrt{hred} & si \ \varepsilon \leq hred \leq 1 - \varepsilon \\ d \cdot (a_1 \cdot hred^3 + b_1 \cdot hred^2 + c_1 \cdot hred + d_1) & si \ -\varepsilon \leq hred \leq \varepsilon \\ 0 & si \ hred \leq -\varepsilon \end{cases} \quad (7)$$

avec les coefficients a_i, b_i, c_i et d_i choisis pour que la fonction soit C^1 .

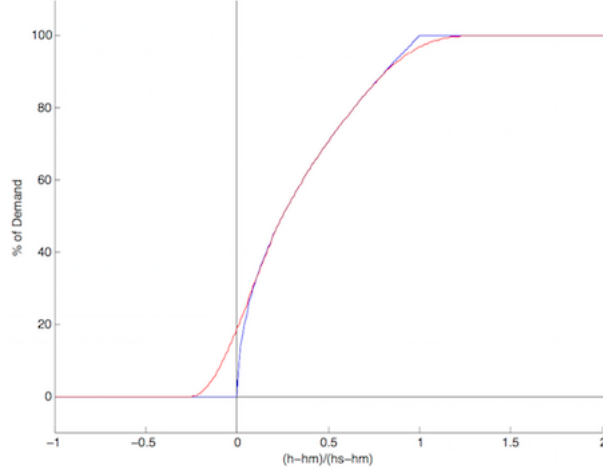


FIGURE 1 – Consommation (% de la demande) en fonction de la pression “réduite” (régularisation de la fonction de Wagner)

Mise en œuvre

Comme lors de l’ajout de la formule de Darcy-Weibach, il a fallu contrôler les utilisations de ce nouveau modèle afin d’adapter le code. Ici encore, le calcul de la Jacobienne sera affecté par cette modification et on se basera sur un des écrits de mon tuteur [5]. Ainsi, une fois le problème d’analyse est mis sous la forme vu Eq.6

Lorsque l’on dérive ce système par rapport à d , on obtient :

$$\begin{cases} A \cdot \partial_d(q) = -C_d - E \cdot \partial_d h \\ D \cdot \partial_d(q) - {}^t A \cdot \partial_d(h) = 0 \end{cases} \quad \text{avec } C_d = \partial_d c \text{ et } E = \partial_h c.$$

Finalement, on exprime, comme précédemment, les termes $\partial_d q$ et $\partial_d h$:

$$\begin{cases} \partial_d q = -D^{-1} \cdot {}^t A \cdot (A \cdot D^{-1} \cdot {}^t A + E)^{-1} \cdot C_d \\ \partial_d h = -(A \cdot D^{-1} \cdot {}^t A + E)^{-1} \cdot C_d \end{cases} \quad (8)$$

où les matrices C_d et E dépendent de la formule choisie pour modéliser la consommation c .

On réalise les mêmes opérations en dérivant par rapport à la rugosité :

$$\begin{cases} A \cdot \partial_{rug}(q) = -E \cdot \partial_{rug} h \\ D \cdot \partial_{rug}(q) - {}^t A \cdot \partial_{rug}(h) = 0 \end{cases} \quad \text{pour obtenir les termes } \partial_{rug} q \text{ et } \partial_{rug} h :$$

$$\begin{cases} \partial_{rug} q = -D^{-1} \cdot B \cdot R_{rug} + D^{-1} \cdot {}^t A \cdot (A \cdot D^{-1} \cdot {}^t A + E)^{-1} \cdot A \cdot D^{-1} \cdot B \cdot R_{rug} \\ \partial_{rug} h = (A \cdot D^{-1} \cdot {}^t A + E)^{-1} \cdot A \cdot D^{-1} \cdot B \cdot R_{rug} \end{cases} \quad (9)$$

On remarquera qu’on ne dérive plus par rapport au diamètre. En effet, la dérivation par rapport au diamètre est très semblable à celle par rapport à la rugosité et a donc été supprimée afin d’alléger l’écriture. On a toujours une jacobienne s’écrivant sous la forme vu en annexe B.2

3.4 Transport

Théorie

L'ajout de constituants dans un réseau d'eau potable peut être bénéfique (ajout de chlore pour assainir) ou néfaste (contamination accidentelle ou volontaire entraînant une diminution de la qualité). La partie du code correspondant au transport est donc un élément très important qui a pour but de déterminer l'origine d'une contamination (où et quand le contaminant a été injecté) ainsi que ses répercussions (quelles zones sont maintenant affectées par cette contamination). Comme précédemment, c'est l'identification de paramètres (ou calage) qui nous intéresse et nous allons donc, dans un premier temps, écrire les équations décrivant ce transport (d'après [6]). On se placera dans le cas du transport du chlore dans une conduite de longueur infinie ; En considérant que la diffusion est négligée devant le transport (ce qui est souvent le cas dans la littérature) et la longueur des conduites est très supérieure à leur diamètre dans une équation de transport-réaction monodimensionnelle, on peut considérer le problème hyperbolique suivant :

$$\begin{cases} \partial_t C(t, x) + u(t) \cdot \partial_x C(t, x) + F(t, x) = 0 \\ C(t = 0, x) = C_0(x) \quad \forall x \in \mathbb{R}^+ \\ C(t, x = 0) = \phi(t) \quad \forall t \geq 0 \end{cases} \quad (10)$$

Où $C(t, x)$ est la valeur de la concentration en chlore, u est la vitesse donnée par le modèle hydraulique, ϕ est la condition aux limites correspondant au profil de chlore à l'extrémité de la conduite et C_0 la condition initiale (résidu de chlore dans la conduite au temps 0).

Enfin, $F(t, x) = kC(t, x)^\alpha$, $\alpha \geq 1$ décrit l'interaction du désinfectant avec le milieu. Le calage de paramètre consiste alors à déterminer l'ordre de réaction α et les classes de coefficients de réaction k . Ici aussi le problème va se résumer à minimiser au sens des moindres carrés la différence entre la solution issue d'un calcul et les mesures pondérée par des poids (ici l'inverse des précisions)

$$P(\theta) = \sum_{i=1}^m \int_{t_0}^{t_f} W_i (C_i(\theta, t) - C_i^{mes})^2 dt \quad (11)$$

où θ est la vecteur contenant les variables à caler (des classes de demandes, de rugosités ou éventuellement de diamètres) et W_i l'inverse des précisions des mesures définies dans la suite et m le nombre de mesure.

On remarquera qu'il faut ajouter une dernière équation qui va traduire la mise à jour de la concentration au noeud aval, c'est-à-dire que cette équation va représenter le mélange éventuel qui se fait en entrée d'une conduite (par exemple lorsque plusieurs conduites arrivent en un même noeud). On prendra une formule traduisant

$$\text{un mélange parfait : } C_{aval} = \frac{\sum_{i \in N_{in}} q_i(t) \cdot C_i}{\sum_{i \in N_{in}} q_i(t)} .$$

Utilisation

Le principal objectif est le calage de paramètres hydrauliques à partir de mesures de concentration pour un réseau d'eau. Une fois le code du transport compris, j'ai

dû l'utiliser afin de pouvoir caler les paramètres hydrauliques. Comme pour le calage effectué précédemment, il a fallu calculer la Jacobienne de la concentration en fonction des paramètres à caler (ici demande et rugosité). On reprend alors l'équation 10 que l'on dérive par rapport à la demande dans un premier temps :

$$\partial_t C_{jD} + u(t) \cdot \partial_x C_{jD} + U_D \partial_x C_j + C_{jD} F_{C_j} = 0 \quad (12)$$

avec $C_{jD} = \partial_D C_j$, $U_D = \partial_D u$ et $F_{C_j} = \partial_{C_j} F$. On remarquera que ici, on travaille avec les grandeurs C_j qui correspondent aux concentrations aux différents noeuds.

Ainsi, la résolution du problème de l'analyse et du transport nous donne $u(t)$ et C_j (et donc $\partial_x C_j$). Il faut ajouter à cela l'équation 8 (ou 4 en DDM) pour nous permettre d'obtenir U_D (cette équation nous donne $\partial_D q$ et on a $u = \frac{q}{S}$). L'objectif étant de caler la demande, on va donc se rapprocher de ce qui était déjà implémenté pour la partie hydraulique, à savoir le schéma suivant : $D^{k+1} = D^k - H^{-1} \cdot \nabla_D P(D)$ où D est la classe de demande à caler, où le calcul de H et $\nabla_D P(D)$ est détaillé en annexe C.2

On peut alors expliciter le schéma de calcul pour actualiser la demande :

$$D^{k+1} = D^k - 2 \left(\int_{t_0}^{t_f} {}^t J \cdot W \cdot J dt \right)^{-1} \int_{t_0}^{t_f} {}^t J \cdot W (C(D, t) - C^{mes}) dt.$$

Enfin, on peut remarquer que l'expression continue de P avec une intégrale peut se discrétiser sous la forme $P(\theta) = \sum_{i=1}^m \sum_{k=1}^{\tau} W_i (C_i(\theta, t_k) - C_{i,k}^{mes})^2$ avec τ le nombre de régime (nombre de discrétisation en temps). L'expression de $\nabla_D P(D)$ est encore plus simple : $\nabla_D P(D) = 2 \cdot {}^t Jac \cdot W (C(D) - C^{mes})$.

Le développement précédent se généralise avec d'autres variables à caler comme la rugosité par exemple (où la jacobienne se notera \sum) pour obtenir une Jacobienne global de la forme suivante :

$$Jac = \begin{pmatrix} \sum_1 & J_1 & & & \\ \sum_2 & & J_2 & & \\ \vdots & & & \ddots & \\ \sum_{\tau} & & & & J_{\tau} \end{pmatrix} \quad (13)$$

matrice définie par bloc où chaque bloc correspond au calcul de la jacobienne en fonction de D pour J et de rug pour \sum .

On pourra remarquer ici que les J_i sont placées de manière à être indépendantes les unes des autres alors que les \sum_i forment une seule colonne de matrices. Cela s'explique par le fait que les classes de demandes dépendent du temps. En effet, il y a plusieurs régimes de consommation (à différentes périodes de la journée, de la semaine ou de l'année) contrairement aux rugosités qui sont toujours les mêmes. Ainsi, la sensibilité par rapport aux classes de rugosité sera toujours calculée de la même façon (avec des débits différents) alors que la sensibilité par rapport aux classes de demande sera différente et indépendante à chaque régime.

4 Résultats et conclusion

4.1 Résultats

Les résultats obtenus seront liés aux différents travaux réalisés. Tout d'abord, regardons les résultats du calage effectué sur les demandes dans le réseau caltest (annexe A.1) après avoir modifié la lecture des mesures afin de vérifier sa fonctionnalité. Les résultats attendus étant 1 et 0.0833, on a les résultats suivants :

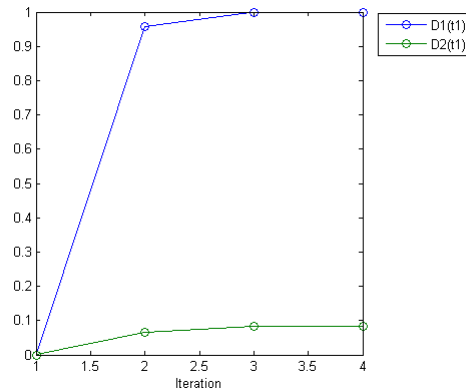


FIGURE 2 – calage de deux classes de demande pour le réseau Caltest

Avec des résidus inférieurs à 10^{-4} ce qui a pu nous permettre de valider cette nouvelle lecture des mesures.

Le réseau test de la partie PDM sera le réseau z0000012emap (annexe A.2) pour lequel on constate que environ 70% des noeuds sont concernés. On présente premièrement les résultats du calcul hydraulique PDM pour résoudre l'Eq. 6 avec la fonction de Wagner régularisée :

PIPES							
from	to	Flowrate [lps]	Velocity [mps]	Total Headloss [mH2O]	v0 [mps]	q0 [lps]	
1	2	13.6811	1.7419	37.11399	0.30	2.4	
1	4	303.8508	4.2986	19.70435	0.30	22	
2	3	25.0340	0.7969	1.44407	0.30	9.6	
2	5	-46.8750	-0.6631	-0.15460	0.30	22	
3	6	6.8121	0.2168	0.16205	0.30	9.6	
4	5	209.0759	2.9578	17.25504	0.30	22	
4	7	74.7749	2.3802	19.17543	0.30	9.6	
5	6	82.4736	1.1668	1.76072	0.30	22	
5	8	58.0968	1.1835	2.27026	0.30	15	
6	9	89.2857	1.2631	0.50987	0.30	22	
7	8	22.2443	0.3147	0.34987	0.30	22	
8	9	0.7143	0.0101	0.00033	0.30	22	

NODES							
NAME	Hmin [mH2O]	Head [mH2O]	Hs [mH2O]	redHead	c (H) [lps]	%FullDemand	H0 [mH2O]
2	10.00	22.8860	30.00	0.6443	35.5221	71.044	12.54
3	10.00	21.4419	30.00	0.5721	18.2219	60.740	20.94
4	19.00	40.2956	39.00	1.0648	20.0000	100.000	35.29
5	10.00	23.0406	30.00	0.6520	21.6305	72.102	28.12
6	5.00	21.2799	25.00	0.8140	-0.0000	100.000	23.27
7	9.00	21.1202	29.00	0.6060	52.5307	65.663	14.57
8	5.00	20.7703	25.00	0.7885	79.6268	88.474	6.95
9	0.00	20.7700	20.00	1.0385	90.0000	100.000	12.65
1	60.00	60.0000	NaN	resource	0.0000		

Dans la partie concernant les résultats sur les noeuds on voit bien les noeuds dont la demande n'est pas satisfaite à 100%. Dans ces résultats, H_s correspond à la Cote désirée et H_{min} à la Cote sol nécessaires au calcul de la "pression réduite" vue lors de l'introduction de la PDM. Ces résultats sont à mettre en regard avec les résultats en DDM du calcul hydraulique (sans prise en compte de la PDM). Dans ce cas moins complexe, les débits sont plus importants et les pressions aux noeuds sont plus faibles. Cela s'explique facilement puisque avec des consommations plus importantes dans le cas de la DDM, il y a plus de "circulation" d'eau et donc des débits supérieurs tandis que plus l'eau "sort" du réseau (par des consommations plus fortes) moins il sera sous "pression". Ainsi, on peut expliquer les premiers résultats issus de ce calcul hydraulique après avoir ajouté la prise en compte de la PDM dans le code Matlab.

On a ensuite voulu utiliser dans la partie calage du code. Comme on l'a vu précédemment, c'est le calcul de la Jacobienne qui est impacté par la PDM et il a donc fallu comparer les résultats obtenus. Par soucis de synthèse, les résultats concernant les dérivées par rapport aux classes de demande et de rugosité des débits et charges sont disponibles en Annexe E.

Là encore, ces résultats sont en accord avec ce qui pouvait être attendus puisque l'on a des débits et des pressions qui sont moins "sensibles" à la demande dans le cas de la PDM. En effet, il semble plutôt cohérent que des débits ou des pressions (qui sont liés à la consommation) soient moins sensibles à la demande lorsque cette consommation est inférieure à la demande (cas PDM) que lorsque cette consommation est prise égale à la demande (DDM). Pour illustrer cela, on pourrait se placer dans le cas d'un noeud avec une forte demande ayant une faible pression. Dans le cas de la PDM, si il y a un changement de demande, il y aura un changement moins important de la consommation puisque cette dernière correspondra à un faible % de la demande au vue de la pression non suffisante, et les débits et pressions seront alors peu impactés. Dans le cas de la DDM, même avec une faible pression, la demande sera satisfaite et un changement de demande aura un fort impact sur les équations d'équilibre hydraulique et donc sur le calcul des débits et pressions.

La dernière partie concerne le placement des capteurs. En effet, il faut choisir les mesures les plus efficaces pour optimiser notre calcul. Pour cela, on va regarder quels sont les variables (débits ou pressions) qui sont le plus sensibles à des erreurs. En plaçant nos capteurs en ces différents points, on lève alors l'incertitude de cette variable. Pour cela, on prendra par convention que les pressions sont précises à 0.1 mètres d'eau et que les débits sont précis à 33% de la valeur du débit si il est inférieur à 0.1 litres par seconde et 10% si il est supérieur à cette valeur.

On se place dans le cas où on cherche à caler des demandes et on crée la matrice : $\Delta = \text{diag}(\text{précisions})$. On va donc pondérer notre Jacobienne définie par $Jac = \begin{pmatrix} \partial_d q \cdot K_d \\ \partial_d h \cdot K_d \end{pmatrix}$ par l'inverse de ces précisions afin de créer la matrice de décision qui nous servira à sélectionner les valeurs à mesurer. On présente les résultats obtenus

sur notre réseau en PDM :

$$\Delta = \begin{pmatrix} 1.368 & 1 \\ 30.385 & 2 \\ 2.503 & 3 \\ -4.688 & 4 \\ 0.681 & 5 \\ 20.908 & 6 \\ 7.477 & 7 \\ 8.247 & 8 \\ 5.810 & 9 \\ 8.929 & 10 \\ 2.224 & 11 \\ 0.238 & 12 \\ 0.1 & 2 \\ 0.1 & 3 \\ 0.1 & 4 \\ 0.1 & 5 \\ 0.1 & 6 \\ 0.1 & 7 \\ 0.1 & 8 \\ 0.1 & 9 \end{pmatrix} \left. \begin{array}{l} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \right\} \begin{array}{l} \text{Tronçons} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \text{Noeuds} \end{array}, \text{Jac} = \begin{pmatrix} 3.258 \\ -92.556 \\ 68.126 \\ -108.889 \\ 3.301 \\ -3.301 \\ -18.011 \\ 223.566 \\ 311.693 \\ 133.566 \\ 60.853 \\ 66.516 \\ -0.163 \\ -0.126 \\ -0.090 \\ -0.124 \\ -0.125 \\ -0.127 \\ -0.169 \\ -0.119 \end{pmatrix} \text{ et enfin } \Delta^{-1} \cdot \text{Jac} = \begin{pmatrix} 2.381 \\ -3.046 \\ 27.214 \\ 23.230 \\ 4.845 \\ -0.158 \\ -2.409 \\ 27.108 \\ 53.651 \\ 14.959 \\ 27.357 \\ 279.361 \\ -1.634 \\ -1.264 \\ -0.901 \\ -1.242 \\ -1.249 \\ -1.266 \\ -1.693 \\ -1.195 \end{pmatrix}$$

Sur ce réseau, on décide de n'avoir qu'une mesure de débit et trois mesures de pressions. On cherche donc la valeur la plus importante en norme absolue dans les 12 premières lignes de la dernière matrice et les 3 plus grandes en norme dans les 8 dernières lignes. Ainsi, on conclut avec ce calcul que pour caler le plus efficacement les demandes à partir d'une mesure de débit et de trois mesures de pression, on mesurera le débit du tronçon 12 (0.714 litres par seconde) et les pressions aux noeuds 2 (12.886 mètres d'eau), 7 (12.12 mètres d'eau) et 8 (15.77 mètres d'eau). Ces mesures sont prises à partir des résultats du calcul hydraulique. Une fois ces mesures entrées dans la fichier Porteau, on a pu effectuer le calage de la demande. Ainsi, on obtient les résultats suivants avec à gauche la classe de demande en litre par seconde et à droite les quatre résidus liés au débit puis aux pressions :

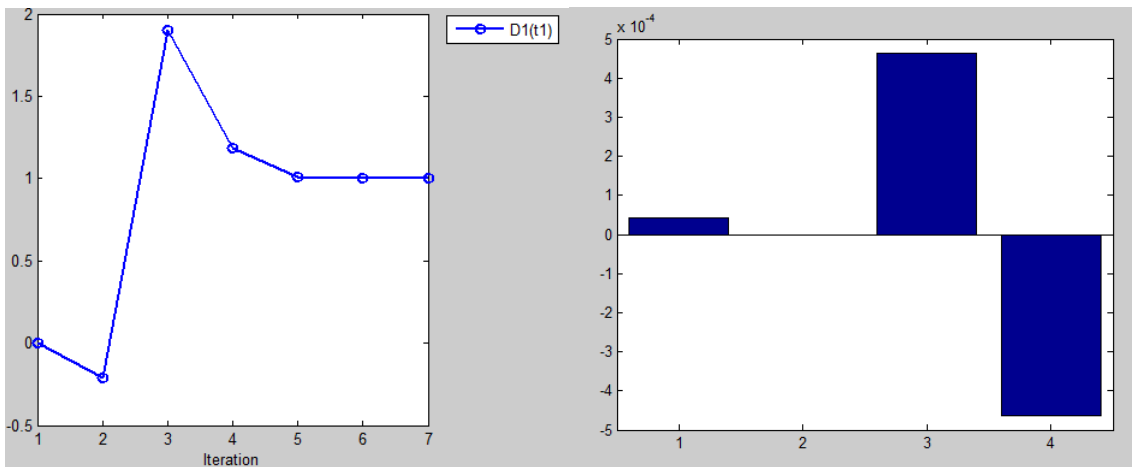


FIGURE 3 – Calage de la demande

On observe alors que la demande est calée à 1 avec des résidus de l'ordre de 10^{-4} ce qui nous permet de valider ce calage. On a également tracé la fonction P que l'on a cherché à minimiser pour confirmer nos résultats. La première itération est surprenante puisque elle ne semble pas aller dans le sens d'une minimisation de cette fonction. Cependant, après avoir zoomé sur cette partie, on a pu constater qu'il y a des parties très "raides" proche de 0 malgré l'apparence "lisse" de la courbe ce qui justifie cette première itération qui nous donne une demande négative. L'algorithme converge donc en 5 itérations et minimise bien la fonction P lorsque le paramètre (ici la demande) est égale à 1 même si ces parties raides semblent indiquer des instabilités du code proche de zéro (annexe C.3). Prendre 0.1 comme valeur initiale pour la demande corrige ce phénomène et permet un très bon calage :

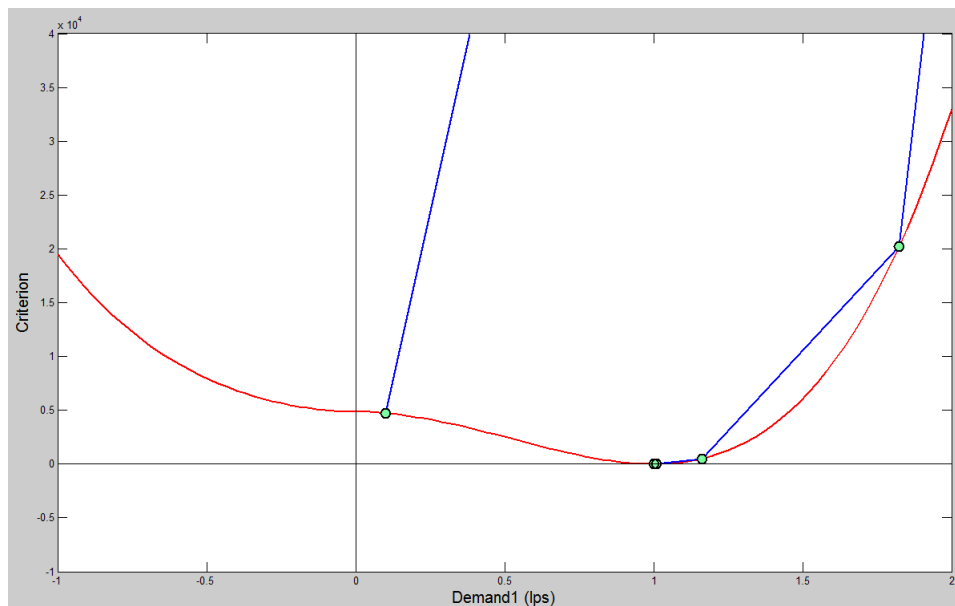


FIGURE 4 – Fonction P

On conclut donc sur la validité de ce qui a été implémenté puisque l'on retrouve bien le résultat attendu puisque la valeur de la demande rentrée dans Porteau était de 1. On a donc pu réaliser un calcul hydraulique sur ce réseau test avec le modèle PDM, puis on a pu caler la classe de demande de ce réseau en utilisant notre algorithme de placement des mesures et les résultats du calcul hydraulique. Pour finaliser cette partie, il faudrait programmer le calcul des deux matrices explicitées plus haut (E et C_d) afin de pouvoir caler cette classe de demande en PDM.

4.2 Conclusion

Il est facile de résoudre le problème de l'analyse pour un réseau de distribution d'eau lorsque l'on possède les informations nécessaires. Cependant, on se rend compte que la plupart du temps ce n'est pas de calculer les débits ou les pressions qui pose problème mais plutôt la connaissance de notre réseau et notamment les demandes des consommateurs ou les paramètres des conduites (rugosité, diamètres, ...). C'est donc la détermination de ces éléments qui est un enjeu important aujourd'hui et le calage de paramètre, élément central de ce stage, qu'il faut développer et optimiser.

Pour cela, j'ai du reprendre un algorithme de calage de paramètres hydrauliques en temps réel déjà implémenté sous Matlab et y ajouter des fonctionnalités pratiques, comme la lecture des mesures depuis le fichier Porteau (ce qui a simplifié leur utilisation), ou basées sur le modèle avec par exemple l'introduction d'une nouvelle loi de perte de charge plus précise.

Cependant la partie la plus importante et qui nécessitait donc le plus de modification résidait dans la définition même du problème puisqu'il a fallu ajouter une dépendance en pression à la consommation. Comme on a pu le voir, cette modification a apporté des changements dans les résultats et a donc aidé à une meilleure prédiction.

Le problème du placement des capteurs est lui aussi impacté par cette modification. En effet, les résultats des Jacobiennes étant différents, on pourrait très bien imaginer que l'algorithme qui calcule le placement des capteurs (détaillé dans la partie résultats) puisse avoir des issues différentes dans les deux cas bien que sur notre exemple les résultats en PDM et en DDM furent identiques.

Enfin, j'ai également pu travailler sur la partie transport qui est en développement et qui a pour but de caler les demandes et rugosités non plus à l'aide de mesures de débits et/ou pressions mais à partir de mesures de concentrations d'un constituant dans le réseaux.

La prochaine étape de ce projet sera de terminer le calage de paramètres hydrauliques à partir de mesures de concentrations et de l'intégrer au code de calcul afin de pouvoir caler ces paramètres avec des mesures mélangeants concentrations, pressions et débits. Ainsi, les entreprises pourront, grâce à Porteau, avoir une solution logicielle qui leur permettra de savoir quels capteurs doivent être placés et à quels endroits afin de pouvoir au mieux caler des paramètres hydrauliques. La finalité sera alors un modèle de simulation fiable du réseau avec un contrôle de la qualité de l'eau, des débits et des pressions. Enfin, la mesure de concentrations en certains points du réseaux pourra permettre de remonter à la source de la contamination afin de pouvoir les limiter lorsqu'elles sont trop importantes et ainsi contrôler les risques et les impacts sur le réseau.

Outre l'autonomie et les compétences en mécanique des fluides qui ont été nécessaires, ce stage m'a principalement permis de développer mes compétences en mathématiques et en programmation. En effet, le calage, qui repose sur un problème d'optimisation, a été amélioré en ajoutant une nouvelle formule de perte de charge ou en introduisant un nouveau modèle pour la consommation. Ces modifications ont été validées par des tests et on a pu comparer des éléments centraux du calage comme par exemple les jacobiennes calculées dans les différents cas. Ce stage m'a également permis de pouvoir m'inscrire dans un groupe de recherche puisque j'ai pu assister à divers réunions concernant les projets de l'entreprise ou les stratégies à adopter au niveau du logiciel Porteau. Pouvoir intervenir dans ces réunions, en comprendre les enjeux et connaître l'utilisation de mon travail était stimulant et intéressant mais également très formateur.

Bibliographie

[1] Piller O. (1995), "Modélisation du fonctionnement d'un réseau. Analyse hydraulique et choix des mesures pour l'estimation de paramètres", thèse à l'université de Bordeaux, Talence, France.

[2] Latappy C. (2013), "Calage en temps réel de la demande pour un modèle de transport dans un réseau d'eau potable", rapport de stage de 2ème année ENSEIRB-MATMECA.

[3] Elhay S. (2014) "Notes on Dunlop transitional flow and convexity", non publié.

[4] Piller O., Jakobus E. Van Zyl and Gilbert D. (2010), "Dual Calibration for coupled flow and transport models of water distribution systems", unité de recherche REBX (Irstea), Cestas, 33000, FRANCE et Département d'ingénierie civile de l'université de Cape Town, Rondebosch, 7701 SOUTH AFRICA.

[5] Piller O. (2014), "Derivation of the sensitivity matrices for DDM and PDM cases", non publié.

[6] Gancel G. (2006), "Modélisation d'un problème inverse pour la qualité de l'eau dans les réseaux d'eau potable", these à l'université de Bordeaux, Talence, France.

[7] Wagner, J. M., Shamir, U., and Marks, D. H. (1988). "Water Distribution Reliability : Simulations Methods." *Journal of Water Resources Planning and Management*, 114(3), 276-294.

[8] Piller, O., Brémond, B., and Poulton, M. (2003). "Least Action Principles Appropriate to Pressure Driven Models of Pipe Networks." *World Water and Environmental Resources Congress (EWRI03)*, Philadelphia (PA), US, printed by ASCE, (CD-ROM).

Annexe A.1

Le réseau caltest, basé sur un réseau réel, se compose de 16 arcs, 15 nœuds simples et 2 réservoirs. Les arcs ont trois rugosités (le coefficient d'Hazen-Williams vaut 106 pour les arcs rouges, 116 pour les bleus et 136 pour le marron). On connaît alors les demandes et les rugosités, ce qui nous permet de vérifier si les résultats du calage réalisé par le code à l'aide de quelques mesures sont corrects.

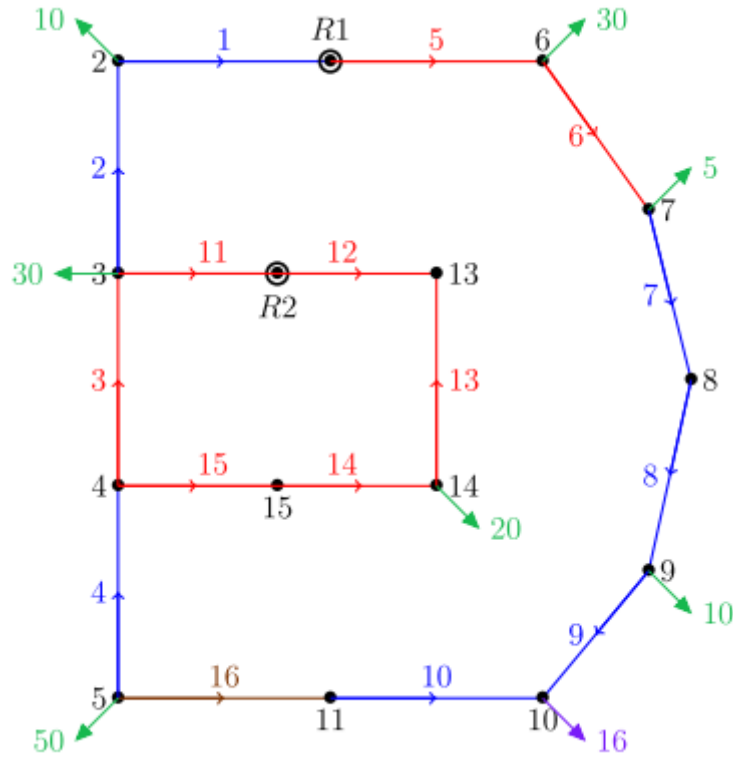


FIGURE 5 – Graphe du réseau Caltest

Le réseau z0000012emap, réseau qui est utilisé pour la PDM, se compose de 12 arcs, 8 nœuds simples et 1 réservoir.

Annexe A.2

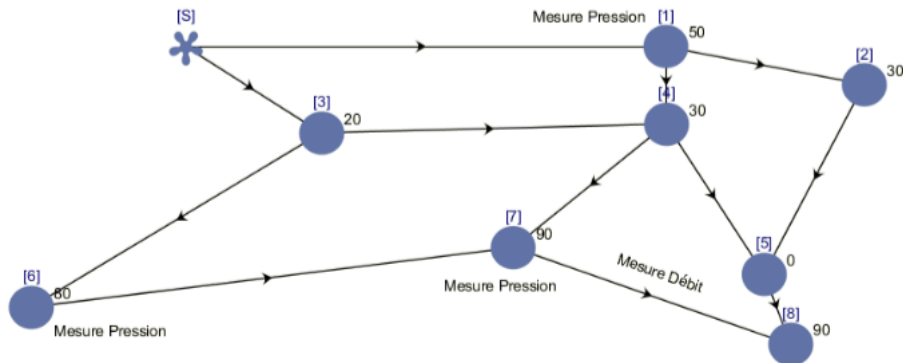


FIGURE 6 – Graphe du réseau z0000012emap

Annexe B.1 Description des grandeurs

pour les cas transitoire et turbulent, on a : $u = u(q, diam, rug) = \frac{rug}{3.7 \cdot diam} + \frac{x}{Re^y}$
 et $(x, y) = (5.74, 0.9)$,

k	α_k	β_k
0	$\frac{5}{(\chi^2 \cdot \ln^2(\hat{u}))}$	$\frac{\tau}{(\chi^3 \cdot \ln^3(\hat{u}))}$
1	$0.128 - \frac{12}{(\chi^2 \cdot \ln^2(\hat{u}))}$	$-\frac{5\tau}{2(\chi^3 \cdot \ln^3(\hat{u}))}$
2	$-0.128 + \frac{9}{(\chi^2 \cdot \ln^2(\hat{u}))}$	$-\frac{2\tau}{(\chi^3 \cdot \ln^3(\hat{u}))}$
3	$0.032 - \frac{2}{(\chi^2 \cdot \ln^2(\hat{u}))}$	$-\frac{\tau}{2(\chi^3 \cdot \ln^3(\hat{u}))}$

avec $\hat{u} = \hat{u}(diam, rug) = \frac{rug}{3.7 \cdot diam} + \frac{x}{4000^y}$, $\tau = 0.00514215$, $\chi = -0.86859$ et $\eta = Re/2000$.

Annexe B.2 Expression de la jacobienne

$$Jac(y)(x) = \begin{pmatrix} S_q \cdot \partial_{rug} q \cdot K_C & S_q \cdot \partial_{diam} q \cdot K_\phi & S_q \cdot \partial_d q \cdot K_d \\ S_h \cdot \partial_{rug} h \cdot K_C & S_h \cdot \partial_{diam} h \cdot K_\phi & S_h \cdot \partial_d h \cdot K_d \\ 0_{md,nc} & 0_{md,n\phi} & S_q \cdot K_d \\ -S_f \cdot A_f \cdot \partial_{rug} q \cdot K_C & -S_f \cdot A_f \cdot \partial_{diam} q \cdot K_\phi & -S_f \cdot A_f \cdot \partial_d q \cdot K_d \end{pmatrix}$$

Où les $S_q, S_h, S_d, S_f, K_C, K_\phi$ et K_d sont les matrices de sélection des mesures et des classes.

Annexe B.3 Consommation en fonction de la pression

Voici les deux autres fonctions utilisées pour la consommation dans le cas de la PDM :

- Polynôme d'ordre 5

$$c(d, h) \begin{cases} d & \text{si } hred \geq 1 \\ d \cdot (hred^3 \cdot (10 - hred \cdot (15 - 6 \cdot hred))) & \text{si } 0 \leq hred \leq 1 \\ 0 & \text{sinon} \end{cases}$$

- Polynôme d'ordre 3

$$c(d, h) \begin{cases} d & \text{si } hred \geq 1 \\ d \cdot (hred^2 \cdot (3 - 2 \cdot hred)) & \text{si } 0 \leq hred \leq 1 \\ 0 & \text{sinon} \end{cases}$$

Annexe C.1 l'algorithme de Levenberg-Marquardt :

Algorithme de calage dont la formule d'itération est la suivante :

$$x^{k+1} = x^k - ({}^t J_k \cdot J_k + e_k S_k^2)^{-1} \cdot {}^t J_k \cdot (y(x^k) - y^{mes})$$

avec $S_{ii}^2 = ({}^t J \cdot J)_{ii}^2 + \phi$ et e_k une suite de réels positifs.

Annexe C.2 Calcul de la hessienne et du gradient

H la hessienne calculée par $H = {}^t C_{jD} \cdot W \cdot C_{jD}$

On sait que $P(D+h) - P(D) = P'(D)(h) + o(h)$, calculons $P(D+h) - P(D)$:

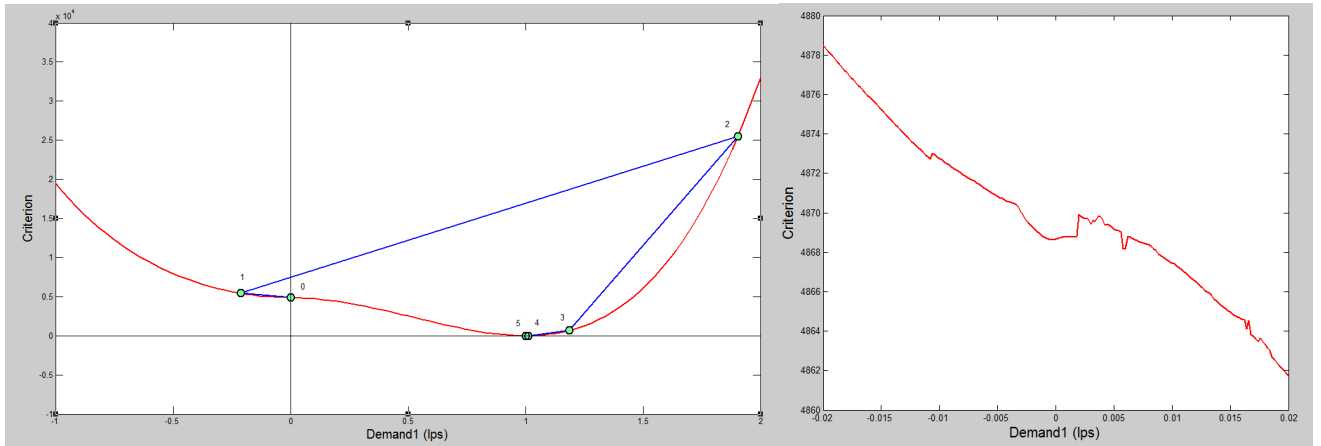
$$\begin{aligned} P(D+h) - P(D) &= \sum_{i=1}^m \int_{t_0}^{t_f} W_i ((C_i(D+h, t) - C_i^{mes})^2 - (C_i(D, t) - C_i^{mes})^2) dt \\ &= 2 \sum_{i=1}^m \int_{t_0}^{t_f} W_i \langle C_{jD}, h \rangle (C_i(D, t) - C_i^{mes}) dt \\ &= \langle 2 \sum_{i=1}^m \int_{t_0}^{t_f} W_i (C_i(D, t) - C_i^{mes}) \cdot C_{jD} dt, h \rangle \end{aligned}$$

On en déduit alors que $\nabla_D P(D) = 2 \sum_{i=1}^m \int_{t_0}^{t_f} W_i (C_i(D, t) - C_i^{mes}) \cdot C_{jD} dt$ et en posant

$$J = \left(\frac{\partial C_i}{\partial D_j} \right)_{ij} \text{ on a l'écriture matricielle : } \nabla_D P(D) = \int_{t_0}^{t_f} 2 \cdot {}^t J \cdot W (C(D, t) - C^{mes}) dt$$

Annexe C.3

Fonction P avec valeur initiale nulle pour la demande (à gauche) et instabilités proche de 0 (à droite) :



Annexe D Expression des matrices R_{rug} , R_ϕ , D et B .

Nous allons exprimer ces quatre matrices dans le cadre de la formule de Darcy-Weibach afin de pouvoir assembler la jacobienne utilisée dans le calage (voir annexe B algorithme de Levenberg-Marquardt). Pour cela, on va reprendre, pour chaque régime, les formulations définies dans la partie 3.2 Ajoute de la formule de perte de charge de Darcy-Weibach :

— Régime laminaire : $D = \partial_q(\xi)$, $D = \frac{128 \cdot \nu \cdot L}{\pi \cdot g \cdot diam^4}$ et donc $D^{-1} = \frac{\pi \cdot g \cdot diam^4}{128 \cdot \nu \cdot L}$.

$$B = \partial_r(\xi), \boxed{B = q}. \quad R_{rug} = \partial_{rug}(r), \boxed{R_{rug} = 0}. \quad R_\phi = \partial_{diam}(r), \boxed{R_\phi = \frac{-4r}{diam}}.$$

— Régime transitoire : $D = \partial_q(\xi)$, $D = \begin{cases} \frac{8L}{\pi^2 \cdot g \cdot diam^5} | q | \sum_{k=0}^3 (\alpha_k + \frac{\beta_k}{u}) \cdot (2\eta^k + q \partial_q \eta^k) \\ + \frac{8L}{\pi^2 \cdot g \cdot diam^5} | q | q \sum_{k=0}^3 \left(\partial_q \alpha_k + \frac{u \cdot \partial_q \beta_k - \beta_k \cdot \partial_q u}{u^2} \right) \cdot \eta^k \end{cases}$

$$B = \partial_r(\xi), \boxed{B = q \cdot | q |}.$$

$$R_{rug} = \partial_{rug}(r), \boxed{R_{rug} = \frac{8L}{\pi^2 \cdot g \cdot diam^5} | q | q \sum_{k=0}^3 \left(\partial_{rug} \alpha_k + \frac{u \cdot \partial_{rug} \beta_k - \beta_k \cdot \partial_{rug} u}{u^2} \right) \cdot \eta^k}$$

$$R_\phi = \partial_{diam}(r), \boxed{R_\phi = \begin{cases} \frac{8L}{\pi^2 \cdot g \cdot diam^5} | q | q \sum_{k=0}^3 \left(\partial_{diam} \alpha_k + \frac{u \cdot \partial_{diam} \beta_k - \beta_k \cdot \partial_{diam} u}{u^2} \right) \cdot \eta^k \\ + \frac{8L}{\pi^2 \cdot g \cdot diam^5} | q | q \sum_{k=0}^3 (\alpha_k + \frac{\beta_k}{u}) \cdot \left(-\frac{5 \cdot \eta^k}{diam} + \partial_{diam}(\eta^k) \right) \end{cases}}$$

— Régime turbulent : $D = \partial_q(\xi)$, $D = \frac{2 \cdot \ln^2(10) \cdot L}{\pi \cdot g \cdot diam^5} \cdot \frac{2|q|}{\ln^2(u)} \left(1 + \frac{y}{\ln(u)} \right)$. $B = \partial_r(\xi)$, $\boxed{B = q \cdot | q |}$.

$$R_{rug} = \partial_{rug}(r), \boxed{R_{rug} = \frac{-2r \cdot \partial_{rug} u}{u \cdot \ln(u)}}. \quad R_\phi = \partial_{diam}(r), \boxed{R_\phi = r \cdot \left(\frac{-2 \partial_{diam} u}{u \cdot \ln(u)} - \frac{5}{diam} \right)}.$$

Annexe E Matrices jacobiennes

Matrices Jacobiennes du débit en fonction des classes de demande :

$$\text{PDM} : J = \partial_{dq} \cdot K_d = \begin{pmatrix} 3.258, -92.556, 68.126, -108.889, 3.301, -3.301, \\ -18.011, 223.566, 311.693, 133.566, 60.853, 66.516 \end{pmatrix}$$

$$\text{DDM} : J = \partial_{dq} \cdot K_d = \begin{pmatrix} 3.956, -115.979, 84.812, -151.720, 9.518, -9.518, \\ -20.070, 281.238, 386.049, 191.238, 69.930, 90.553 \end{pmatrix}$$

Matrices Jacobiennes de la pression en fonction des classes de demande :

$$\text{PDM} : J = \partial_{dqh} \cdot K_d = (-0.163, -0.126, -0.090, -0.124, -0.125, -0.127, -0.169, -0.119)$$

$$\text{DDM} : J = \partial_{dqh} \cdot K_d = (-0.237, -0.182, -0.128, -0.178, -0.178, -0.184, -0.242, -0.171)$$

Matrices Jacobiennes du débit en fonction des classes de rugosité (à gauche en PDM et à droite en DDM) :

$$\begin{pmatrix} 13.476 & 1.874 & 0.414 & -0.326 \\ 11.531 & 43.636 & -61.244 & -0.390 \\ -3.457 & -17.385 & 18.349 & 0.160 \\ 9.618 & -157.039 & 20.169 & -0.415 \\ -0.274 & -142.171 & 1.245 & 0.012 \\ 0.274 & 142.171 & -1.245 & -0.012 \\ 1.864 & -41.043 & 61.849 & -0.066 \\ -9.881 & 14.893 & -18.917 & 0.461 \\ -13.338 & -2.492 & -0.568 & 0.621 \\ -9.881 & 14.893 & -18.917 & 0.461 \\ 1.909 & -41.439 & 61.729 & 0.008 \\ -1.557 & 183.268 & -63.072 & 0.148 \end{pmatrix} \begin{pmatrix} 13.476 & 1.874 & 0.414 & -0.326 \\ 11.531 & 43.636 & -61.244 & -0.390 \\ -3.457 & -17.385 & 18.349 & 0.160 \\ 9.618 & -157.039 & 20.169 & -0.415 \\ -0.274 & -142.171 & 1.245 & 0.012 \\ 0.274 & 142.171 & -1.245 & -0.012 \\ 1.864 & -41.043 & 61.849 & -0.066 \\ -9.881 & 14.893 & -18.917 & 0.461 \\ -13.338 & -2.492 & -0.568 & 0.621 \\ -9.881 & 14.893 & -18.917 & 0.461 \\ 1.909 & -41.439 & 61.729 & 0.008 \\ -1.557 & 183.268 & -63.072 & 0.148 \end{pmatrix}$$

Matrices Jacobiennes des pressions en fonction des classes de rugosité (à gauche en PDM et à droite en DDM) :

$$\begin{pmatrix} 0.010 & -0.094 & -0.021 & 0.016 \\ 0.006 & 0.010 & 0.004 & 0.017 \\ 0.004 & 0.001 & 1E-4 & 0.017 \\ 0.006 & 0.012 & 0.003 & 0.017 \\ 0.006 & 0.043 & 0.005 & 0.017 \\ 0.006 & 0.009 & 0.004 & 0.017 \\ 0.010 & -0.088 & -0.027 & 0.016 \\ 0.005 & 0.010 & 0.003 & 0.017 \end{pmatrix} \begin{pmatrix} 0.015 & -0.141 & -0.032 & 0.022 \\ 0.008 & 0.012 & 0.005 & 0.022 \\ 0.006 & 0.001 & 1E-4 & 0.023 \\ 0.008 & 0.017 & 0.005 & 0.022 \\ 0.008 & 0.025 & 0.005 & 0.022 \\ 0.008 & 0.005 & 0.007 & 0.022 \\ 0.015 & -0.137 & -0.036 & 0.022 \\ 0.007 & 0.013 & 0.004 & 0.022 \end{pmatrix}$$

Résumé

Le projet *SMaRT – Online^{WDN}*, sur la fiabilité et la sécurité des réseaux d'eau, est en lien avec le développement et l'amélioration du code Matlab du logiciel Porteau. Le calage, qui repose sur un problème d'optimisation, a été amélioré en ajoutant une nouvelle formule de perte de charge (Darcy-Weisbach) qui tient compte du régime de l'écoulement et en introduisant un nouveau modèle avec une consommation dépendante de la pression. A ces modifications, il faut également ajouter l'utilisation des sensibilités pour le placement optimal de capteurs afin de minimiser l'impact des imprécisions sur l'estimation des paramètres. Sur des exemples précis, il est montré l'utilité du modèle PDM pour caler la demande. Enfin, j'ai aussi travaillé sur le problème du transport où le calage de paramètres hydrauliques à partir de mesures de concentration est initié. Ceci devra être achevé pour que cette partie soit intégrée au logiciel Porteau.

Abstract

The *SMaRT – Online^{WDN}* project on reliability and safety of water systems is linked with the development and improvement of Porteau software Matlab code. A calibration method, based on a least squares problem formulation, was developed for an improved hydraulic model; a new head loss formula (Darcy-Weisbach) is added that considers the flow regime; and a new model with a consumption dependent of the available pressure is introduced. Furthermore, sensibility coefficients were used to decide optimal placement of sensors to minimize the effect of inaccuracies for least square estimations. On particular examples, It was shown that calibration of a PDM problem leads to a more physical estimation of demand. Finally, I also worked on the problem of transport where calibration of hydraulic parameters from concentration measurements is intended. Future work will be to integrate the latter development in the Porteau software.